

Community (UM)

Opensource Software

[GNU General Public License GPL GNU Definition Open Source Initiative](#)

Opensource Cooperation

These notes are much more important in an open community like OpenCPN. Otherwise work issues tend to be stored out of sight and we risk two or more are working on the same thing clueless of each others. Open cooperation and problem solving communication is a better way.

There is a wide range of skills & skill levels in this community. We all try to help each other, but we need to understand that this is all voluntary, and intended to be fun. We all share the common interest of supporting and creating great Chart Plotter Navigation software.

Participation

There are many ways to participate in and assist this Open Source Project. Programming skills are advantageous but not necessary because many other different skills and efforts are needed. See where you can help!

User Participation

Language Translator: Assist by [Crowdin Translation of OpenCPN and Plugins](#)

User Manual Editor: Assist by [Editing](#)

User Feedback: Suggestions, User Interface Suggestions, Feature Requests, Bugs should be done exclusively via:

- [OpenCPN Cruiser's Forum](#) Find or make an appropriate thread For discussion.
- [Tracker](#) Sometimes called Flyspray. You must register first.

Note:

1. Please refrain from inserting your comments, bugs & feature requests into *Alpha* and *Beta* Testing.
2. This distracts the programmers from their goal, disrupts communications and the points, as good as they may be are generally lost!
3. Programmers are often working on much bigger projects than the User Interface, which may be associated with support of many OS, migration to newer tools, or introducing big new features..

4. Let the programmers work in peace on tough problems.

Beta Tester

Beta Testing is conducted in a Real Time environment (not using Debug) by volunteer testers or end users when the development team determines the software is ready.

Any user can be a beta tester, they need to know how to install and use the software and report. Using CI tools is inappropriate for Beta Testing.

Programming Education

We unfortunately don't have manpower to teach people programming and duplicate every piece of documentation out there.

Dave has suggested these websites for Training in C++ which is the most basic skill required. In any case they should be bookmarked and studied.

- [LearnCpp](#)
- [CSplusplus](#)
- [Github Opensource Getting Started](#)

We also do not have energy, time, resources, whatever, to maintain the outdated toolchain, dependencies and know-how.

However all the build instructions are updated for the current master and that is what developers should use. Version 4.8.2 is history. (Pavel -Nohal)

Alpha Tester

Done early on by skilled testers at the specific request of programmers who are actively creating big new features or internal optimizations. These skilled testers are asked by the programmers to help achieve the main goals of the program to get it ready for Beta Testing, using a "Debug" environment. They do not "Bug" the programmers with:

1. Additional feature requests.
2. UI changes.
3. Known Bugs (the programmers already know about)
4. Other trivia.

They debug the program using their software tools and report the actions and error codes using MSVC++ call stack, or Linux debug tools. Beta testing can only begin when the development team is sure there are no obvious bugs such as null pointer deref or other errors that can be caught through testing inside a debug environment. Beta testing begins when the development team wishes it.

Early Alpha Tester

Sometimes called “heat-seekers”. For this type of tester, using the CI products for informal alpha testing of unreleased code is very useful to the devs. The main benefit to devs is the rapid cross-platform exercise of the app, where testers bang away on the interface using their favorite workflows, and attempt to break it. Most devs don't keep current images of all 5 platforms available for extended testing, and in any case this takes a lot of time. However with the software tools below, the time spent compiling is dramatically less.

It simply must be understood that this alpha testing is ad-hoc, and is not tracked in any persistent manner. But it does advance the code-base integrity quickly.

However do not “bug” the programmers. Simply report your findings as completely as possible using the Github “Issues” section. Include any pertinent logs, screenshots or files. Do not combine issues into one post and identify the program version or date.

Programmer

Programming skills vary greatly. Some are learning, some have many years of experience. Respect each other and work together when possible.

Software Tools

Debugging see the “Compiling” pages“

- [Linux Compile & Debug - GNU Data Display Debugger Webpage Intro to DDD Manual](#)
- [Windows Compile & Debug - Windows Debuggers](#)
- [Compile & Debug on MacOS](#)

Online Repositories - Free to Opensource

- [Github Opensource Help Getting Started OpenCPN Github](#) Plugin Repositories
- [Sourceforge OpenCPN](#) Older compiled OpenCPN
- [Sourceforge Packaging Data](#) Necessary for Compiling
- [Sourceforge Older Plugins](#) Older Plugins

Additional Software Tools which integrate with and push to Github - Free to OpenSource :

- [Travis CI](#) “Build apps with confidence”
- [Appveyor](#) “Continuous Integration solution for Windows” [Docs](#)

1. Programmers use these tools to setup automatic builds of the current github “head”. This involves having certain *.yml files in the main directory and requires the plugin author to have an account and enable the feature. Linux and Windows executable become available for download and testing.
2. OpenCPN and some plugins have TravisCI and Appveyor enabled.
3. Alpha Testers working closely with Programmers can then access and download the most recent

“build” without having to compile.

4. These tools are used for ad hoc “Alpha Testing”.
5. OpenCPN Programmers do not use these tools for Continuous Integration (CI) or Continuous Releases (CR) or Test Driven Development (TDD).

See [Advanced Debugging Tips](#)

From:
<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:community

Last update: **2021/09/23 00:54**

